y2522li_a1q1

January 29, 2021

1 A1

[1]:

```
# Standard imports
import numpy as np
np.seterr(all='ignore');  # allows floating-point exceptions
import matplotlib.pyplot as plt
```

1.1 Q1: randfp

[2]:

```
def randfp(t, L, U):
    '''
     b = randfp(t, L, U)

     Generate a random normalized binary floating-point number with
     t digits, and an exponent in the range [L,U]. For example,

       b = randfp(5, -4, 4)

     might yield

       b = '-0.10111b-2'

     or

       b = '+0.11100b4'

     Note that the output is a string, and that the first character is
     always either a '+' or '-'. The number after the 'b' is
     the exponent for the base 2, although the exponent itself is
     represented in base-10. For example,

       b = '+0.11100b4'

     represents the number 0.11100 x 2^4.
    '''
    from random import seed
    from random import randint

    res = ''

    a = randint(0, 1)
    if a == 1:
        res = res + '+'
    else:
        res = res + '-'
    res = res + '0.1'

    for _ in range(t-1):
        a = randint(0, 1)
        res = res + str(a)

    res = res + 'b'
    a = randint(L, U)
    res = res + str(a)

    return res
```

[3]:

```
b = randfp(5, -4, 4)
print(b)
```

```
+0.10001b-4
```

[4]:

```
? randfp
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: